Testing Euclidean Minimum Spanning Trees in the Plane [*] [†]

Artur Czumaj
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ 07102, USA
czumaj@cis.njit.edu

Christian Sohler
Heinz Nixdorf Institute
Computer Science Department
University of Paderborn
D-33102 Paderborn, Germany
csohler@upb.de

Martin Ziegler
Heinz Nixdorf Institute
Computer Science Department
University of Paderborn
D-33102 Paderborn, Germany
ziegler@uni-paderborn.de

Abstract

Given a Euclidean graph G over a set P of n points in the plane, we are interested in verifying whether G is an Euclidean minimum spanning tree (EMST) of P or G differs from it in more than $\epsilon\, n$ edges. We assume that G is given in adjacency list representation and the point/vertex set P is given in an array. We present a property testing algorithm that accepts graph G if it is an EMST of P and that rejects with probability at least $\frac{2}{3}$ if G differs from every EMST of P in more than $\epsilon\, n$ edges. Our algorithm runs in $O(\sqrt{n/\epsilon} \cdot \log^2(n/\epsilon))$ time and has a query complexity of $O(\sqrt{n/\epsilon} \cdot \log(n/\epsilon))$.

Keywords: Euclidean minimum spanning tree, property testing, randomized algorithms.

[*]The results presented in this paper appeared in a preliminary form as a part of the paper "Property testing in computational geometry," in *Proceedings of the 8th Annual European Symposium on Algorithms (ESA'00)*, pages 155–166, volume 1879 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2000 [6].

[†]Research supported in part by NSF grants ITR-CCR-0313219 and CCR-0105701, and by DFG grant Me 872/8-2.

1 Introduction

The problem of finding a minimum spanning tree in a graph belongs to one of the most fundamental problems in algorithmic graph theory. In this paper we study a relaxation of this problem for the class of geometric (Euclidean) graphs. We investigate the problem of testing whether a given graph G is a minimum spanning tree for a set of points P in the plane or it is far from any minimum spanning tree for P.

We consider a set P of n points in the Euclidean plane $\mathbb{R}^2$ and a geometric graph $G = (P, E)$ with vertex set P and edge set E. Graph $G = (P, E)$ is called a *Euclidean minimum spanning tree (EMST)* of point set P if G is a minimum spanning tree of the complete Euclidean graph of P. The complete Euclidean graph is a complete weighted graph where each edge $e = (p, q) \in P \times P$ has weight equal to the Euclidean distance between p and q. For simplicity, we make a standard assumption in computational geometry that P is in general position, i.e., all edge weights are distinct. In this case it is known that the EMST is unique (see, e.g., [9]). We assume that G is given in adjacency list representation and the set P is given in an array.

The main result of this paper is a property testing algorithm that for a given P and G accepts the input if G is the EMST of P and that rejects with probability at least $2/3$ every graph G that differs from the EMST of P in more than $\epsilon\, n$ edges. Our algorithm runs in $O(\sqrt{n/\epsilon} \cdot \log^2(n/\epsilon))$ time and has query complexity of $O(\sqrt{n/\epsilon} \cdot \log(n/\epsilon))$.

Notice that since the complexity of finding the EMST for a set of n points in $\mathbb{R}^2$ is $\Theta(n \log n)$ (even if an approximate solution is sought), our result provides a $\widetilde{\mathcal{O}}(\sqrt{n})$-time, thus sublinear-time approximation procedure to test if a given geometric graph is the EMST.

1.1 Related research

Our result lies on the intersection of classical optimization and property testing. We study the classical optimization minimum spanning tree problem and our goal is to approximately verify if a given graph is a minimum spanning tree. Our result follow the framework of *property testing* [12, 23], which is the computational task to decide if a given object satisfies a certain predetermined property (in our case, input graph is a minimum spanning tree) or is far from every object having this property. If the input object neither has the property nor is far from it, then the algorithm may answer arbitrarily. Thus, the outcome of a property testing algorithm can be seen as an approximation of a property of the input.

The main reason of increasing popularity of property testing in recent years (see, e.g., surveys [10, 11, 22]) is that for a variety of problems the framework of property testing can lead to *sublinear-time* algorithms, that is algorithms that can approximately verify if an object has a given property without the need to examine the whole object. This has some potential applications in massive data sets and other situations, when even reading the input might be prohibitively expensive. In a sequence of papers, various property testing algorithms have been developed for a variety of problems, starting with graph problems through string problems to problems on matrices, see the survey works [10, 11, 22] and the references therein.

In this paper we present for the first time a property testing algorithm for the *minimum spanning tree (MST)* problem. The problem of finding an MST is one of the most fundamental and most extensively studied problems in algorithms. For arbitrary graphs, the currently fastest deterministic algorithm due to Chazelle runs in time $\mathcal{O}(n + m\,\alpha(m, n))$ [2], where n is the size of the vertex set and m is the number of edges in the input graph (see also [21]). Karger et al. [16] gave a linear-time randomized algorithm. The problem of *verifying* if the input graph is a minimum spanning tree of another graph in the general case has been investigated in a series of papers [8, 17, 19], where $\mathcal{O}(n + m)$-time algorithms have been given. A better situation is known for Euclidean graphs in $\mathbb{R}^d$. In the case considered in the current paper, that is for $d = 2$ (on the plane), Shamos and Hoey [24] gave an $\mathcal{O}(n \log n)$-time algorithm for finding an EMST (notice that in this case $m = \Theta(n^2)$).

Recently, (after publishing the preliminary version [6] of the current paper) three sublinear-time approximation algorithms have been presented for the problem of estimating the *weight* of the MST. These algorithms are designed in a similar flavor as that in property testing. Chazelle et al. [3] presented an algorithm that, given a connected graph in adjacency list representation with average degree D, edge weights in the range $[1 \dots W]$, and a parameter $0 < \varepsilon < \frac{1}{2}$, approximates, with high probability, the weight of a MST in time $\widetilde{\mathcal{O}}(D\, W\, \varepsilon^{-3})$ within a factor of $(1 + \varepsilon)$. Czumaj et al. [4] gave a $\widetilde{\mathcal{O}}(\sqrt{n} \cdot \mathrm{poly}(1/\varepsilon))$-time for a similar problem for geometric graphs, but this algorithm is assuming that the input graph is provided on a top of some additional geometric data structures. Recently, Czumaj and Sohler [5] obtained an $\widetilde{\mathcal{O}}(n \cdot \mathrm{poly}(1/\varepsilon))$-time algorithm that estimates the weight of the MST in any metric graph to within a factor of $(1 + \varepsilon)$.

1.2 Outline

After we introduce some basic notation in Section 2, in Section 3 we develop a property tester for disjointness of geometric objects, which is used as a subroutine in the EMST tester. Next, in Section 4, we present our property testing algorithm for the EMST problem.

2 Preliminaries

We start with some basic definitions needed in this section before we discuss the input representation:

Definition 2.1 *A* geometric graph *for* P *is a weighted graph* $G = (P, E)$ *with vertex set* P *and edge set* $E \subseteq P \times P$ *(the edges can be interpreted as straight-line segments connecting the endpoints). The weight of an edge* (p, q) *is implicitly given by the Euclidean distance between* p *and* q *in* $\mathbb{R}^2$.

Definition 2.2 *A geometric graph for* P *that is the minimum spanning tree of the complete geometric graph for* P *is called the* Euclidean minimum spanning tree (EMST) *of* P.

Next we define when a graph is "far" from the EMST. Typically, a distance measure for graph properties depends on the number of entries in the graph representation that must be changed to obtain a graph that has the tested property.

Definition 2.3 *Let* $G = (P, E)$ *be a geometric graph for* P *and let* $\mathbb{T} = (P, \mathbb{E})$ *be the Euclidean minimum spanning tree of* P. *We say* G *is* ϵ-far from being the Euclidean minimum spanning tree of P *(or, in short,* ϵ-far from EMST*) if one has to modify (insert or delete) more than* ϵn *edges in* G *to obtain* $\mathbb{T}$*, that is :*

$$|E \setminus \mathbb{E}| + |\mathbb{E} \setminus E| \quad > \quad \epsilon n \ .$$

Input representation. We assume that both the point set and the graph are given as an oracle, and the point set is represented by a function $f : [n] \to \mathbb{R}^2$ (here and throughout the paper we will use the notation $[n] := \{1, \dots, n\}$ for the set of integer numbers between 1 and n). The algorithm may query the oracle for the value of $f(i)$ for some $i \in [n]$ and gets in return the position of the i-th point of P.

The geometric graph is given in the *unbounded length adjacency list representation* introduced in [20]. The unbounded length adjacency list model is a general model for sparse graphs. The graph structure is represented by adjacency lists of varying length. Our property tester may query for the degree $\mathrm{deg}(p)$ of a vertex p and for each $i \le deg(p)$ it may query for the i-th neighbor of p. We represent the vertex set of our graph by the set of numbers $[n]$. Thus we can easily obtain the position of a vertex p from the point set representation by querying for the value of $f(p)$.

The goal of property testing is to develop efficient *property testers*. A property tester for EMSTs is an algorithm that gets a distance parameter ϵ and the size n of the input point set P (which equals the number of nodes of G). The property tester has *oracle access* to the function f representing the point set and to the graph G. A property tester is an algorithm that:

- *accepts* G, if G is the EMST of P, and
- *rejects* G with probability at least $\frac{2}{3}$, if G is ϵ-far from Π.

[1] If G is neither the EMST nor ϵ-far from it, then the outcome of the algorithm can go either way.

Complexity of property testers. There are two types of possible complexity measures for property testers: The *query complexity* and the *running time*. The query complexity measures the number of queries to the oracle asked by a property testing algorithm. If one counts also the time the algorithm needs to perform other tasks than querying the input function values, then the obtained complexity is called the *running time* of the property tester.

3 Disjointness of geometric objects

Before we start working on the EMST problem we first consider a simpler problem of disjointness of geometric objects, which will be useful in our analysis of the EMST problem. Let $\mathbb{O} = \{O_1, \dots, O_n\}$ be a set of arbitrary geometric objects, each being an (implicitly represented) subset of $\mathbb{R}^d$. Our goal is to test if all objects in $\mathbb{O}$ are pairwise *disjoint*. Two geometric objects O_i and O_j are said to be *disjoint* if $O_i \cap O_j = \emptyset$. A set $\mathbb{O} = \{O_1, \dots, O_n\}$ of n geometric objects is *pairwise disjoint* if each pair of objects O_i and O_j is disjoint, $1 \le i, j \le n$.

We represent sets of geometric objects by a function $\varphi : [n] \to \mathcal{R}$ where $\mathcal{R}$ contains implicit representations of all geometric objects of a certain class of objects, e.g. all line segments, points, or rectangle. For example, when we consider sets of line segments in the $\mathbb{R}^d$ then $\mathcal{R} = \mathbb{R}^d \times \mathbb{R}^d$. The property tester for disjointness of geometric objects is used later when we design a property tester for the Euclidean minimum spanning tree.

Definition 3.1 *A set $\mathbb{O}$ of n objects in the $\mathbb{R}^d$ is ϵ-far from being* pairwise disjoint *if one has to remove more than ϵn objects from $\mathbb{O}$ to obtain a disjoint set of objects.*

Testing algorithm. We consider the following property tester for disjointness of geometric objects.

DISJOINTNESSTESTER(set of arbitrary geometric objects $\mathbb{O}$)
Choose a set $S \subseteq \mathbb{O}$ of size $8\sqrt{n/\epsilon}$ uniformly at random
if S is disjoint **then** *accept*
else *reject*

[1]We consider a *one-sided error* model, though in the literature also a *two-sided error* model has been considered, see [10, 11, 22].

Theorem 1 *Algorithm* DISJOINTNESSTESTER *is a property tester for disjointness of geometric objects. Its query complexity is* $\mathcal{O}(\sqrt{n/\epsilon})$ *and its running time is* $T(8\sqrt{n/\epsilon}) + \mathcal{O}(1)$*, where* $T(m)$ *is the time to decide whether a set of* m *input objects is disjoint.*

Proof : We have to prove that (1) algorithm DISJOINTNESSTESTER accepts every set of disjoint geometric objects and (2) that it rejects every set of geometric objects that is ϵ-far from disjoint with probability at least $2/3$.

Part (1) is immediate: If $\mathbb{O}$ is pairwise disjoint, the every subset of $\mathbb{O}$ is also pairwise disjoint and so algorithm DISJOINTNESSTESTER accepts $\mathbb{O}$.

Thus let us suppose that $\mathbb{O}$ is ϵ-far from disjoint and prove part (2). It is easy to see that we can apply $k = \epsilon n/2$ times the following procedure to $\mathbb{O}$: pick a pair of intersecting objects W_i, $i \in [k]$, and remove it from $\mathbb{O}$. In order to prove that DISJOINTNESSTESTER is a property tester it is sufficient to show that with probability at least $2/3$ at least one of these pairs is in the sample set S. We apply Lemma A.1 [7] (see Appendix A) and standard amplification arguments. We consider the sample set S as four independently selected sets $S_1^*, S_2^*, S_3^*, S_4^*$, each of size $\frac{2n}{(2k)^{1/2}}$ and then apply Lemma A.1 to obtain:

$$\mathbf{Pr}\left[\exists j \in [k] : (W_j \subseteq S)\right] \;\geq\; 1 - \prod_{1 \le i \le 4} \left(1 - \mathbf{Pr}\left[\exists j \in [k] : (W_j \subseteq S_i^*)\right]\right) \;\geq\; 1 - (3/4)^4 \;\geq\; 2/3 \enspace .$$

□

4 Testing Euclidean minimum spanning trees

We begin with a simple claim that states some basic properties of Euclidean minimum spanning trees (see, e.g., [9, 18]):

Claim 4.1 *Every Euclidean minimum spanning tree of a point set in general position in* $\mathbb{R}^2$ *has maximum degree less than or equal to five, is connected, and its straight-line embedding is crossing-free.* □

Now we want to introduce some additional notation that will be useful to simplify the description of the algorithm and its analysis.

Definition 4.2 *For a given point set* P*, a geometric graph* $G = (P, E)$*, and the Euclidean minimum spanning tree* $\mathbb{T} = (P, \mathbb{E})$ *of* P*, the* EMST-completion *of* G *is the geometric graph* $G_C = (P, E \cup \mathbb{E})$ *that contains all edges that are in* G *or in* $\mathbb{T}$*.*

In the next subsection we will present a property tester for EMST that works for a special class of input graphs which we call *well-shaped*. The restriction to well-shaped graphs simplifies the analysis of the algorithm and it allows a clear view on the important features of the property tester.

Definition 4.3 *We call a geometric graph* G well-shaped *if*

- *it has maximum degree of 5,*
- *it is connected, and*
- *the straight-line embedding of its EMST-completion is crossing-free.*

Notice that by Claim 4.1, if a geometric graph G is the EMST of P then it is well-shaped. Of course, the reverse is not true. Still, however, we can first test if the input graph G is well-shaped and only if it is, we can test if it is the EMST of P. This suggests the following line of the attack: We first test if the input graph is far from a well-shaped graph. If this is the case then we can reject the graph by Claim 4.1. If the input graph passes the test, then we know that with good probability it is either close to a well-shaped graph or it is well-shaped. If the graph is well-shaped we can use the testing algorithm for the special case of well-shaped graphs that will be presented in Sections 4.1–4.3. Then, in Sections 4.4–4.5, we show that we can relax the assumption that the graph is well-shaped and the algorithm will work also for graphs close to well-shaped. This will establish our algorithm.

4.1 Properties of EMSTs in well-shaped graphs

We now design a property tester for EMST for the case that our input graph is well-shaped. First, we give an overview of the algorithm.

Let G denote a well-shaped geometric graph with vertex set P. We first pick a sample set $S \subseteq P$ using some randomized scheme to be described later. Next, we find the subgraph G_S of G that is induced by the vertex set S. Then we compute the EMST-completion of G_S. If the EMST-completion has a cycle then we reject the input, otherwise we accept.

Now, we proceed with the details. We first show that if the EMST-completion of G_S contains a cycle then we can always reject the input graph. We use the following lemma which follows easily from standard theory of minimum spanning trees (see, e.g., [25, Chapter 6]). (This lemma makes use of the fact that the EMST of P is unique.)

Lemma 4.4 *Let $S \subseteq P$ be a subset of P and let $p, q \in S$. If the edge $e = (p, q)$ does not belong to the EMST of S, then e does not belong to the EMST of P.*

Proof : The proof is by contradiction. Let us suppose that e does not belong to the EMST of S and e belongs to the EMST $\mathbb{T}$ of P. The removal of e in $\mathbb{T}$ cuts $\mathbb{T}$ into two trees. These two trees induce a partition of P into two subsets P_1 and P_2. Since e belongs to the EMST of P, e must also be the shortest edge between these two subsets. Let $S_1 = P_1 \cap S$ and $S_2 = P_2 \cap S$. P_1 and P_2 are not empty since one vertex of e is in each of the sets. Then e is also the shortest edge between S_1 and S_2 and therefore it belongs to the EMST of S; contradiction. $\square$

The following are two immediate consequences of Lemma 4.4 that we will use later in the paper.

Corollary 4.5 *Let G be a geometric graph for P. Let $S \subseteq P$ and let G_S be the subgraph of G induced by S.*

- *If the EMST-completion $G' = (P, E')$ of G contains a cycle [2] $C = (p_0, \ldots, p_k)$ of length $k \geq 3$ with $p_i \in S$ for all $0 \leq i \leq k$, then there is a cycle in the EMST-completion of G_S.*
- *If the EMST-completion of G_S contains a cycle $C = (p_0, \ldots, p_k)$ of length $k \geq 3$, then G is not the EMST of P.* $\square$

Now let us consider an input graph $G = (P, E)$ that is ϵ-far from EMST. Our goal is to design a randomized sampling scheme such that the EMST-completion of the subgraph of G induced by the sample

[2] Here, $C = (p_0, \ldots, p_k)$ is a cycle (of length k) if $p_i \in P$ for all $i \in [k]$, $p_0 = p_k$, $(p_{i-1}, p_i) \in E'$ for all $i \in [k]$ and $p_i \neq p_j$ for all $i, j \in [k]$, $i \neq j$.

set contains a cycle with high probability. Let $\mathbb{T} = (P, \mathbb{E})$ be the EMST of P and let $G_C = (P, \mathbb{E} \cup E)$ be the EMST-completion of G. In the following we refer with *red* edges to the edges in $\mathbb{E} \setminus E$ and with blue edges to the edges in E. (Notice a fundamental difference between red and blue edges, in that red edges are given *implicitly* only, since they do not belong to the input graph G.) We will show that in our analysis, it is sufficient to focus on "short" cycles that contain at most two red edges.

Definition 4.6 *Let C be a cycle of length k in the EMST-completion of G. We call C ϵ-*short *if (1) it is of length k, where $k \le \frac{72}{\epsilon}$ and (2) it contains at most two red edges.*

In our algorithm we try to find ϵ-short cycles that satisfy some additional "topological" properties. We will exploit the fact that G is well-shaped, in particular, that the EMST-completion of G has a crossing-free straight-line embedding. Hence we will use a topological representation of the geometric graph G to exploit the fact that every minimal cycle in a (well-shaped) planar geometric graph corresponds to a face in its straight-line embedding. In order to use this approach in a formal framework we will consider the geometric graph G not only as an undirected graph, but at the same time also using its "directed" representation by "replacing" each undirected edge (p, q) by two directed edges $[p, q\rangle$ and $[q, p\rangle$.

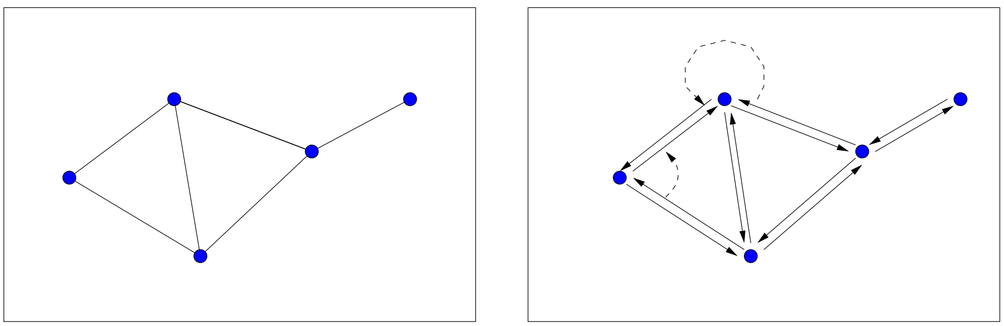

Figure 1: A straight-line embedding and its planar map representation.

For every vertex p in G we (cyclically) sort incident outgoing edges in clockwise order around the vertex p with respect to the Euclidean positions of the edges' endpoints. (This sorting is done only implicitly, but since we assume that each vertex has a constant degree — at most five, each time we consider a vertex we can in constant time sort its incident edges.) The *successor* of a directed edge $[p, q\rangle$ is the edge $[q, r\rangle$ where r is the vertex adjacent to q that precedes p in the adjacency list of q (sorted in clockwise order around q); if q has degree one, then $r = p$. Furthermore, for an edge $e = [p, q\rangle$ in G the k*th successor*, $k \ge 0$, is defined recursively as follows: the 0th successor of $[p, q\rangle$ is $[p, q\rangle$ itself, and for $k > 0$, the kth successor of $[p, q\rangle$ is the successor of the $(k-1)$st successor of $[p, q\rangle$. Similarly, edge e is the *predecessor* of edge e' if edge e' is the successor of edge e, and e is the k*th predecessor* of e' if e' is the kth successor of e. With these definitions, cycles of succeeding edges correspond to faces of the straight-line embedding. Such a representation of G is called a *planar map* for the straight-line embedding of G (see Figure 1). We denote it $\widetilde{G}$.

We have introduced the planar map representation of a graph G because it describes the faces of the corresponding embedding in a simple way (using succeeding edges in $\widetilde{G}$). We observe that the correspondence between the faces in the embedding of G and the cycles of succeeding edges in $\widetilde{G}$ is one to one. We also note that each (directed) edge is contained in exactly one cycle of succeeding edges.

Let G be a well-shaped geometric graph for P and let $C = (p_0, \dots, p_k)$ be a cycle in the planar map of the EMST-completion of G. Then C is called *topological* if for every two consecutive edges on the cycle $[p_i, p_{i+1}\rangle$ and $[p_{i+1}, p_{i+2}\rangle$, $[p_{i+1}, p_{i+2}\rangle$ is the successor of $[p_i, p_{i+1}\rangle$. We also call the corresponding cycle in G topological.

The following *key lemma* shows that every well-shaped geometric graph that is far from EMST must contain many short topological cycles in its EMST-completion.

Lemma 4.7 *Let* $G = (P, E)$ *be a well-shaped geometric graph for* P. *If* G *is* ϵ*-far from EMST, then there are at least* $\frac{\epsilon n}{100}$ ϵ*-short topological cycles in the EMST-completion of* G.

Proof : Let $\mathbb{T} = (P, \mathbb{E})$ be the EMST of P. Let $E_B = E \setminus \mathbb{E}$ denote the blue edges in the EMST-completion of G and let $E_R = \mathbb{E} \setminus E$ denote the red edges in the EMST-completion of G. Since G is ϵ-far from EMST, we have $|E_B| + |E_R| > \epsilon n$ by definition.

Now, let H denote the EMST-completion of G and let $\widetilde{H}$ denote the planar map of its straight-line embedding. Let ρ denote the number of faces in the straight-line embedding of H. Then, $\widetilde{H}$ has ρ disjoint topological cycles since each face is bounded by a unique cycle of succeeding edges. Since H is planar, connected, and has more than $n - 1 + \epsilon n/2$ edges, we apply Euler's formula to deduce that $\rho \geq \epsilon n/2$.

Now, let $s(f)$ denote the number of (directed) edges in the topological cycle bounding face f. Since by Euler's formula $\sum_f s(f) \leq 6n$, there can be at most $\epsilon n/8 \leq \frac{\rho}{4}$ faces f with $s(f) \geq \frac{48}{\epsilon}$. Therefore, there are at least $\frac{3\rho}{4}$ faces f with $s(f) < \frac{48}{\epsilon}$.

Since $|E_R| \leq \rho$, the number of directed red edges is at most 2ρ. Hence, the number of topological cycles with 3 or more red edges can be at most $\frac{2\rho}{3}$. Since we have shown that there are at least $\frac{3\rho}{4}$ topological cycles having less than $\frac{48}{\epsilon}$ edges, at least $\frac{\rho}{12} \geq \frac{\epsilon n}{24}$ of them have at most two red edges. □

Let G be a well-shaped geometric graph for P. For every vertex $p \in P$, we define its *topological* k*-neighborhood* as the set of vertices that are the endpoints of the edges that are either the ith successor of any edge incident to p, $0 \leq i \leq k$, or the jth predecessor of any edge incident to p, $0 \leq j \leq k$. The topological k-neighborhood of a vertex p is denoted $\mathcal{N}_G^{top}(p, k)$.

The following claim follows from the fact that every well-shaped graph has maximum degree of 5.

Claim 4.8 *Let* G *be a well-shaped geometric graph for* P. *For every vertex* $p \in P$, *we can find its topological* k*-neighborhood in time* $\mathcal{O}(k)$. □

4.2 Simple property tester in well-shaped graphs

Now we are ready to present our first property tester for testing if the input well-shaped graph is the EMST of a given input point set. Later, in Section 4.3 in Lemma 4.18, we present a faster, but more complex algorithm.

Our first approach is to sample uniformly at random a sufficiently large set Q of points in P. Then we add to the sample set the topological $\frac{72}{\epsilon}$-neighborhood of every point in Q. Provided that the set Q is sufficiently large, we prove in Lemma 4.12 that if G is ϵ-far from EMST, then the obtained set of vertices will contain a certain ϵ-short topological cycle in the EMST-completion of G with probability at least $2/3$. By Corollary 4.5, this would certify that G is not an EMST.

We assume that G is well-shaped. Notice first that every ϵ-short topological cycle either

1. is a cycle consisting of at most $\frac{72}{\epsilon}$ blue edges, or
2. is a path consisting of at most $\frac{72}{\epsilon}$ blue edges whose two endpoints are connected by a red edge, or

3. is a path consisting of at most $\frac{72}{\epsilon}$ blue edges whose two endpoints are connected by a path consisting of exactly two red edges, or

4. consists of two paths containing at most $\frac{72}{\epsilon}$ blue edges whose endpoints are connected to each other by two red edges.

We first observe that if there are many ϵ-short topological cycles of type (1) or (2), then we can easily spot them.

Lemma 4.9 *Let* $G = (P, E)$ *be a well-shaped geometric graph. If the EMST-completion of* G *contains at least* $\frac{\epsilon n}{200}$ ϵ*-short topological cycles of type (1) or (2), then a set* $Q \subseteq P$ *of size* $4000/\epsilon$ *chosen uniformly at random with probability at least* $\frac{2}{3}$ *contains at least one vertex from a* ϵ*-short topological cycle.*

Proof : Since G is well-shaped, it has a maximum degree of 5 and therefore the EMST-completion of G has maximum degree of at most 10. Thus, every vertex $p \in P$ is contained in at most 10 ϵ-short topological cycles. This implies that the set P_C of all vertices that are contained in at least one ϵ-short topological cycle (of type (1) or (2)) has cardinality at least $\frac{\epsilon n}{2000}$. Now, if we choose a set $Q \subseteq P$ of size $\frac{4000}{\epsilon}$ taken uniformly at random from P, then

$$\mathbf{Pr}\left[Q \cap P_C = \emptyset\right] \ \leq \ \left(1 - \frac{|P_C|}{n}\right)^{|Q|} \ \leq \ \left(1 - \frac{\epsilon}{2000}\right)^{|Q|} \ \leq \ \frac{1}{3} \ .$$

Therefore,

$$\mathbf{Pr}\left[Q \cap P_C \neq \emptyset\right] \ \geq \ \frac{2}{3} \ .$$

□

Next, we observe that for any ϵ-short topological cycle C of type (1) or (2), for every vertex v from C all other vertices from C belong to the topological $\frac{72}{\epsilon}$-neighborhood of v. This motivates us to define the sample set S as the topological $\frac{72}{\epsilon}$-neighborhood of all vertices in Q. Since the set Q contains at least one vertex from any ϵ-short topological cycle of type (1) or (2) with probability at least $2/3$, we can conclude that S contains all vertices from a particular ϵ-short topological cycle of type (1) or (2) with probability at least $\frac{2}{3}$. Since every vertex of the topological cycle is contained in our sample set we know by Corollary 4.5 that the EMST-completion of the subgraph induced by our sample contains a cycle. Thus our property tester rejects the input with probability $\frac{2}{3}$ if it is ϵ-far from EMST and its EMST-completion contains at least $\frac{\epsilon n}{200}$ ϵ-short topological cycles of type (1) or (2).

We can summarize our discussion above in the following lemma.

Lemma 4.10 *Let* $G = (P, E)$ *be a well-shaped geometric graph and let* $Q \subseteq P$ *be a set of size* $4000/\epsilon$ *chosen uniformly at random from* P*. If the EMST-completion of* G *contains at least* $\frac{\epsilon n}{200}$ ϵ*-short topological cycles of type (1) or (2), then the set*

$$S \ = \ \bigcup_{p \in Q} \mathcal{N}_G^{\mathrm{top}}\left(p, \frac{72}{\epsilon}\right)$$

contains all vertices of at least one ϵ*-short topological cycle with probability at least* $2/3$*.* □

The ϵ-short topological cycles of type (3) and (4) are more difficult to detect. However, we can still use a very similar approach as for cycles of type (1) or (2), but this time we must find two vertices that belong to the same ϵ-short topological cycle. Suppose that in the EMST-completion of G there are at least $\frac{\epsilon n}{200}$ ϵ-short topological cycles of type (3) or (4). As before, we first take a random subset Q of P, but this time the size of Q is $\Theta(\sqrt{n/\epsilon})$. Then, we define the sample set S to be the union of the topological $\frac{72}{\epsilon}$-neighborhood of all vertices in Q. We show now that the so defined sample set is sufficient to certify that G (if it is ϵ-far from EMST) is not an EMST by proving an analogous statement to Lemma 4.10 for cycles of type (3) and (4):

Lemma 4.11 *Let* $G = (P, E)$ *be a well-shaped geometric graph and let* $Q \subseteq P$ *be a set of size* $80\sqrt{n/\epsilon}$ *chosen uniformly at random from* P*. If the EMST-completion of* G *contains at least* $\frac{\epsilon n}{200}$ ϵ*-short topological cycles of type (3) or (4), then the set*

$$S = \bigcup_{p \in Q} \mathcal{N}_G^{top}\left(p, \frac{72}{\epsilon}\right)$$

contains all vertices of at least one ϵ*-short topological cycle with probability at least* $2/3$*.*

Proof : For every ϵ-short topological cycle C of type (3) let us define the set W_C to contain two vertices: one vertex on the blue path in C and the vertex incident to the two red edges in C. Similarly, for every ϵ-short topological cycle C of type (4) let us define the set W_C to contain one vertex from the first blue path in C and one vertex from the second blue path in C.

Since each vertex $p \in P$ belongs to at most 10 ϵ-short topological cycles we can select from the sets W_C the sets W_i, $1 \le i \le \frac{\epsilon n}{2000}$, such that the sets W_i are disjoint and for each i, $1 \le i \le \frac{\epsilon n}{2000}$, there is an ϵ-short topological cycle C with $W_i = W_C$. Then we apply Lemma A.1 [7] with $k = \frac{\epsilon n}{2000}$, $\ell = 2$, and $s = \frac{20n}{\sqrt{\epsilon n}} = 20\sqrt{n/\epsilon}$ and obtain

$$\mathbf{Pr}\left[\exists j \in [k] : (W_j \subseteq Q)\right] \ \ge\ 1 - (1 - 1/4)^4 \ \ge\ \frac{2}{3}\ .$$

Now observe that if $W_C \subseteq Q$ then all vertices of a cycle C are in S. Therefore, the lemma follows. □

Now we are ready to prove that the following algorithm is a property tester for EMST:

```
EMST-TEST-SIMPLE(G, ϵ)
    s = 80√(n/ϵ) + 4000/ϵ
    choose a set Q ⊆ P of size s uniformly at random
    S = ⋃_{q∈Q} N_G^top(q, 72/ϵ)
    compute the subgraph G_S induced by S
    compute the EMST-completion G_C of G_S
    if G_C contains a cycle then reject
    else accept
```

Lemma 4.12 *Let* G *be a well-shaped geometric graph for* P*. Then there is a property tester that in time* $\mathcal{O}\left(\sqrt{n/\epsilon^3} \cdot \log(n/\epsilon)\right)$ *and with query complexity* $\mathcal{O}(\sqrt{n/\epsilon^3})$ *accepts the input if* G *is an EMST of* P *and rejects the input with probability at least* $\frac{2}{3}$ *if* G *is* ϵ*-far from EMST.*

Proof : By Corollary 4.5, if the input graph $G = (P, E)$ is the EMST then EMST-TEST-SIMPLE accepts.

Now let us consider the case when G is ϵ-far from EMST. Then by Lemma 4.7 we know that there are $\epsilon n/100$ ϵ-short topological cycles in the EMST completion of G. It follows that there are $\epsilon n/200$ cycles of type (1) and (2) or $\epsilon n/200$ cycles of type (3) or (4). By Lemma 4.10 and 4.11 we know that the sample taken by EMST-TEST-SIMPLE contains an ϵ-short topological cycle with probability at least $2/3$. By Corollary 4.5 we know that then there is a cycle in the EMST-completion of the subgraph induced by our sample. Hence the algorithm rejects in such a case.

The query complexity of the algorithm is immediate. Its running time follows from Claim 4.8 and the fact that the EMST completion of a graph with m vertices can be computed in time $\mathcal{O}(m \log m)$. □

4.3 Improved property tester in well-shaped graphs

In this section we present a modification of the property tester EMST-TEST-SIMPLE(G, ϵ) that has a slightly better complexity. In our analysis above we were always trying to catch one initially fixed single vertex from each blue path although an ϵ-short topological cycle can contain as many as $\frac{72}{\epsilon}$ vertices. We now want to take the length of the topological cycles into consideration. Furthermore, we were always taking topological $\frac{72}{\epsilon}$-neighborhoods of all vertices. This strategy should be applied to the cycles that have as many as $\frac{72}{\epsilon}$ edges, but it is not necessary for shorter cycles. Our approach now is to improve the complexity of the property tester by combining these two observations. We show that if the input graph is well-shaped then the following algorithm is a property tester for EMST:

```
EMSTTEST(G, ε)
    s = 1700 √(n/ε) + 192000/ε + 4000/ε
    S = FINDCYCLE(G, s, ε)
    compute the subgraph G_S induced by S
    compute the EMST-completion G_C of G_S
    if G_C contains a cycle then reject
    else accept
```

Where the procedure FINDCYCLE is the following:

```
FINDCYCLE(G, s, ε)
    S^(0) = ∅
    for i = 1 to 2s do
        j = 0
        pick a vertex p^(i) ∈ P uniformly at random
        while j ≤ log 72/ε do
            j = j + 1
            flip a coin
            if head then exit
        S^(i) = S^(i−1) ∪ N_G^top(p^(i), 2^j)
    return S^(2s)
```

First of all, we observe that by Corollary 4.5 algorithm EMSTTEST accepts every Euclidean minimum spanning tree. Therefore, we only have to prove that if the input graph is ϵ-far from EMST then it is rejected with probability at least $2/3$.

Let us assume that G is well-shaped and ϵ-far from EMST. Then, by Lemma 4.7, there are at least $\frac{\epsilon n}{100}$ ϵ-short topological cycles in the EMST-completion of G. Let $\mathfrak{C}_j$, $j = 1, 2, 3, 4$, denote the set of all ϵ-short topological cycles of type (j) in the EMST-completion of G. Now we consider separately cycles in $\mathfrak{C}_1 \cup \mathfrak{C}_2$ and cycles in $\mathfrak{C}_3 \cup \mathfrak{C}_4$. By our discussion above we have either $|\mathfrak{C}_1 \cup \mathfrak{C}_2| \geq \frac{\epsilon n}{200}$ or $|\mathfrak{C}_3 \cup \mathfrak{C}_4| \geq \frac{\epsilon n}{200}$.

Cycles of type (1) and (2). Suppose that G is a geometric graph for P with maximum degree 5 and there are at least $\varepsilon\, n$ ϵ-short topological cycles of type (1) or (2) in the EMST-completion of G for $\varepsilon = \frac{\epsilon}{200}$. We first consider the probability that a fixed ϵ-short topological cycle $C \in \mathfrak{C}_1 \cup \mathfrak{C}_2$ is contained in the sample set. Let ℓ denote the number of vertices in cycle C. Then the probability that in round i of the FINDCYCLE procedure vertex $p^{(i)}$ is one of the ℓ vertices of cycle C is $\frac{\ell}{n}$. Furthermore, the probability that the topological neighborhood of $p^{(i)}$ is chosen large enough to contain all vertices of C is at least $\frac{1}{2\ell}$. Overall, for a fixed cycle C the probability that a vertex of C is chosen in round i and that the topological neighborhood of the vertex is large enough is at least $\frac{1}{2n}$. If the cycles are vertex disjoint then it is simple to prove that after $\mathcal{O}(\frac{1}{\epsilon})$ rounds at least one cycle is completely contained in the sample set with constant probability. Unfortunately, in the general case the cycles are not vertex disjoint. To overcome this technical problem we use the planar map representation of G and the following trick for the analysis: Instead of taking the whole topological 2^j-neighborhood of vertex $p^{(i)}$ we assume that our algorithm selects only one of the outgoing edges (in its planar map representation) uniformly at random. Then it includes only the 2^j successors and predecessors of the chosen edge in the planar map representation of G. Clearly, this procedure considers only a subset of the vertices considered in the original procedure. Nevertheless, we can show that the set of vertices we pick using this procedure is still sufficiently large. We can now use the fact that the (directed) cycles are edge disjoint. Assume that we pick a vertex that belongs to a cycle C. Provided that the chosen neighborhood is large enough we still have to choose the correct outgoing edge to have all vertices of C in the sample set. Since our graph has a degree bound of 5 the probability that this edge is chosen is at least $1/5$. Since type (2) cycles consist of a path of $\ell - 1$ (directed) blue edges the probability that $p^{(i)}$ is one of the $\ell - 1$ origins of these edges is $\frac{\ell-1}{n} \geq \frac{\ell}{2n}$ (a directed edge points from its *origin* to its *destination*). Hence the probability that a cycle C of type (1) or (2) is completely contained in the sample set is at least $\frac{1}{20n}$. We know that the cycles are disjoint and so the probability that at least one cycle is completely contained in the sample set in round i is at least $\frac{\varepsilon n}{20n} = \frac{\varepsilon}{20}$. Let X_C denote the indicator random variable for the event that cycle $C \in \mathfrak{C}_1 \cup \mathfrak{C}_2$ is completely contained in the sample set. Then we have for $s \geq 20/\varepsilon = 4000/\epsilon$:

$$\mathbf{Pr}\left[\forall C \in \mathfrak{C}_1 \cup \mathfrak{C}_2 : X_C = 0\right] \;\leq\; \left(1 - \frac{\varepsilon}{20}\right)^{2s} \;\leq\; \frac{1}{3}$$

and hence

$$\mathbf{Pr}\left[\exists C \in \mathfrak{C}_1 \cup \mathfrak{C}_2 : X_C = 1\right] \;\geq\; \frac{2}{3}$$

and so we have just proved:

Lemma 4.13 *Let* G *be a geometric graph for* P *with maximum degree* 5 *that has at least* $\varepsilon n = \frac{\epsilon n}{200}$ *topological* ϵ*-short cycles of type (1) or (2). If algorithm* FINDCYCLE(G, s, ϵ) *is invoked with* $s \geq 4000/\epsilon$ *then the set* $S^{(2s)}$ *returned by the algorithm contains an* ϵ*-short topological cycle with probability at least* $\frac{2}{3}$. □

Cycles of type (3) and (4). Let G be a geometric graph with maximum degree 5. Let us further assume that there are at least $\varepsilon\, n$ topological ϵ-short cycles of type (3) and (4) in the EMST-completion of G, for

$\varepsilon = \frac{\epsilon}{200}$. We show that the sample set computed by algorithm FINDCYCLE contains every vertex of at least one ϵ-short topological cycle with good probability.

Recall that cycles of type (4) consist of 2 paths of blue edges connected by two red edges. Cycles of type (3) are a special case of type (4) cycles: The shorter path has length 0. For each cycle $C \in \mathfrak{C}_3 \cup \mathfrak{C}_4$ let $X_C^{(i)}$ denote the indicator random variable for the event that all vertices of the longer (blue) path of cycle C are in $S^{(i)}$. Let $Y_C^{(i)}$ be the indicator random variable for the event that all vertices of the shorter (blue) path of cycle C are in $S^{(i)}$. Furthermore, let $\Delta^{(i+1)}$ be the indicator random variable for the event that there is a cycle $C' \in \mathfrak{C}_3 \cup \mathfrak{C}_4$ with $X_{C'}^{(i)} = 0$ and $X_{C'}^{(i+1)} = 1$. We say that a cycle $C \in \mathfrak{C}_3 \cup \mathfrak{C}_4$ is *half-contained* in $S^{(i)}$ if $X_C^{(i)} = 1$. Cycle C is *contained* in $S^{(i)}$ if $X_C^{(i)} = 1$ and $Y_C^{(i)} = 1$.

We analyze the algorithm in two steps. We first show that with high probability many (at least $\varepsilon\, s/80$) topological ϵ-short cycles are half-contained in the set $S^{(s)}$. Then we show that the set $S^{(2s)}$ contains at least one cycle $C \in \mathfrak{C}_3 \cup \mathfrak{C}_4$ with high probability.

Claim 4.14 *Let the outcome of the random choices in round* 1 *to* i *of the* **for**-*loop of* FINDCYCLE *be fixed. If*

$$\sum_{C \in \mathfrak{C}_3 \cup \mathfrak{C}_4} X_C^{(i)} < \frac{\varepsilon s}{2} \tag{1}$$

then

$$\mathbf{Pr}\left[\Delta^{(i+1)} = 1\right] \geq \frac{\varepsilon}{40} . \tag{2}$$

Proof : Let us assume that (1) holds. Then we observe that:

$$\sum_{C \in \mathfrak{C}_3 \cup \mathfrak{C}_4} X_C^{(i)} < \frac{\varepsilon\, s}{2} \leq \frac{\varepsilon\, n}{2} ,$$

since $s \leq n$. We conclude that we have more than $\varepsilon\, n/2$ cycles in $\mathfrak{C}_3 \cup \mathfrak{C}_4$ that are not half-contained in $S^{(i)}$. If $p^{(i+1)}$ is one of the vertices of the longer path of one of these cycles and if the topological neighborhood included in FINDCYCLE is large enough then we have $\Delta^{(i+1)} = 1$. To estimate the probability for $\Delta^{(i+1)} = 1$ we apply the same approach as in the analysis for the case of type (1) and (2) cycles. This yields immediately (observing that we have $\varepsilon\, n/2$ cycles instead of $\varepsilon\, n$):

$$\mathbf{Pr}\left[\Delta^{(i+1)} = 1\right] \geq \frac{1}{2\ell} \cdot \frac{\ell}{2n} \cdot \frac{1}{5} \cdot \frac{\varepsilon\, n}{2} = \frac{\varepsilon}{40} .$$

□

Our next goal is to show that there are at least $\varepsilon s/80$ cycles that are half-contained in $S^{(s)}$.

Claim 4.15

$$\mathbf{Pr}\left[\sum_{C \in \mathfrak{C}_3 \cup \mathfrak{C}_4} X_C^{(s)} \leq \varepsilon\, s/80\right] \leq e^{-\varepsilon s/300} .$$

Proof :

$$\mathbf{Pr}\left[\sum_{C\in\mathfrak{C}_4\cup\mathfrak{C}_4} X_C^{(s)} \le \frac{\varepsilon\, s}{80}\right] \;\le\; \mathbf{Pr}\left[\sum_{1\le i\le s} \Delta^{(i)} \le \frac{\varepsilon\, s}{80}\right] \;\le\; \mathbf{Pr}\left[\sum_{1\le i\le s} B^{(i)} \le \frac{\varepsilon\, s}{80}\right] \;,$$

where $B^{(i)}$ are independent 0–1 variables with $\mathbf{Pr}[B^{(i)} = 1] = \varepsilon/40$. The latter inequality follows from Claim 4.14. We now apply a Chernoff bound [15, inequality (7)] to obtain

$$\mathbf{Pr}\left[\sum_{C\in\mathfrak{C}_4\cup\mathfrak{C}_4} X_C^{(s)} \le \frac{\varepsilon\, s}{80}\right] \;\le\; \mathbf{Pr}\left[\sum_{1\le i\le s} B^{(i)} \le \left(1-\frac{1}{2}\right)\cdot\frac{\varepsilon\, s}{40}\right] \;\le\; e^{-\varepsilon\, s/320} \;.$$

□

Let $W^{(i+1)}$ denote the indicator random variable for the event that there exists $C \in \mathfrak{C}_3\cup\mathfrak{C}_4$ with $X_C^{(i)} = 1$ and $Y_C^{(i)} = 0$ and $Y_C^{(i+1)} = 1$.

Claim 4.16 *Let the outcome of the random choices in round* 1 *to* i *of the procedure* FINDCYCLE *be fixed. If*

$$\sum_{C\in\mathfrak{C}_3\cup\mathfrak{C}_4} X_C^{(i)} \;>\; \frac{\varepsilon\, s}{80}$$

then

$$\mathbf{Pr}\left[W^{(i+1)}\right] \;\ge\; \frac{\varepsilon\, s}{1600\, n} \;.$$

Proof : We assume that there are more than $\varepsilon\, s/80$ cycles that are half-contained in $S^{(i)}$. Again, we use essentially the same approach as in the case of type (1) and (2) cycles. We observe that there is a problem with cycles of type (3). Since the length of the shorter path is 0 there is no directed edge in this path. Thus we have to slightly modify our approach. We use the following sampling scheme for the analysis: Instead of taking the whole topological 2^j-neighborhood of $p^{(i)}$ we choose a number k between 1 and 6 uniformly distributed. If k is between 1 and 5 we include the 2^j predecessors and successors of the k-th edge incident to $p^{(i)}$. In the case $k = 6$ we only include the vertex $p^{(i)}$ in the sample. Then we get that the probability that a cycle C is contained in the sample is at least $\frac{1}{2\ell}\cdot\frac{\ell}{2n}\cdot\frac{1}{6} = \frac{1}{24n}$. We have more than $\varepsilon\, s/80$ cycles that are half-contained in $S^{(i)}$. Therefore we obtain that:

$$\mathbf{Pr}\left[W^{(i+1)}\right] \;\ge\; \frac{\varepsilon\, s}{1920\cdot n} \;.$$

□

Lemma 4.17 *Let* G *be a geometric graph for* P *with maximum degree* 5 *that has at least* $\varepsilon\, n = \frac{\epsilon n}{200}$ *topological* ϵ*-short cycles of type (3) or (4). Then* FINDCYCLE *is an algorithm with (expected) query complexity* $\mathcal{O}(\sqrt{n/\epsilon}\log(n/\epsilon))$ *that samples a set* $S \subseteq P$, $|S| \ge 1700\sqrt{n/\epsilon} + 192000/\epsilon$, *such that the EMST-completion of the subgraph* $G_{S^{(2s)}}$ *induced by* $\mathcal{S}^{(2s)}$ *has an* ϵ*-short topological cycle.*

Proof : Let $\varepsilon = \epsilon/200$ and let G be a geometric graph for P with maximum degree 5 that has at least $\varepsilon\, n$ topological ϵ-short cycles of type (3) or (4). By Claim 4.16, the probability that there is a cycle in the EMST-completion of the subgraph induced by $S^{(2s)}$ is greater than or equal to

$$1 - \left(\mathbf{Pr}\left[\frac{1}{s}\sum_{C\in\mathfrak{C}_3\cup\mathfrak{C}_4} X_C^{(s)} \le \frac{\varepsilon}{80}\right] + \left(1-\frac{\varepsilon\, s}{1920\, n}\right)^s\right) \;.$$

Choosing $s \geq 1700\sqrt{n/\epsilon} + 192000/\epsilon$, this bound together with Claim 4.15 for $n \geq 4$ implies that the probability that there is a cycle in the EMST-completion of the subgraph induced by $S^{(2s)}$ is greater than or equal to $1 - (e^{-2} + e^{-5}) \geq \frac{4}{5}$. □

Obtaining deterministic query complexity. It is easy to modify the algorithm such that the query complexity has an upper bound of $\mathcal{O}(\sqrt{n/\epsilon}\log(n/\epsilon))$ by insignificantly increasing the error of the algorithm. We can do this in the following way: We run algorithm EMSTTEST and stop, if it either accepts or rejects or if the sample size becomes too large. Let X_S denote the random variable for the size of $S^{(2s)}$. We stop the algorithm and accept the input if we find out that the size of $S^{(2s)}$ becomes larger than $10 \cdot \mathbf{E}[X_S]$. By Markov inequality we have:

$$\mathbf{Pr}[X_S \geq 10\,\mathbf{E}[X_S]] \leq \frac{1}{10} .$$

Hence it follows that our new algorithm rejects a geometric graph that is ϵ-far from EMST with probability $4/5 - 1/10 \geq 2/3$. Thus it is a property tester with a deterministic bound of $\mathcal{O}(\log(n/\epsilon) \cdot \sqrt{n/\epsilon})$ on the query complexity of our algorithm (rather than expected query complexity).

Lemma 4.18 *Let* G *be a well-shaped geometric graph for* P*. Then there is a property tester that in time* $\mathcal{O}(\log^2(n/\epsilon) \cdot \sqrt{n/\epsilon})$ *and with query complexity of* $\mathcal{O}(\log(n/\epsilon) \cdot \sqrt{n/\epsilon})$ *accepts the input* G *if* G *is an EMST of* P *and that rejects the input with probability at least* $\frac{2}{3}$ *if* G *is* ϵ*-far from EMST.*

Proof : Follows from Lemmas 4.7, 4.13 and 4.17. □

4.4 Property tester in graphs with maximum degree 5

Now we want to remove the condition that the input graphs are well-shaped. In this subsection we do the first step towards that goal. We develop a property tester for connectivity and crossing-free EMST-completions. Then we replace the well-shaped condition for EMSTTEST by the assumption that the input graph has maximum degree 5. Before EMSTTEST is invoked we test if the input graph is $\epsilon/200$-far from connected and if its EMST-completion has a crossing-free straight line embedding. Thus we may assume that EMSTTEST gets an input graph that is $\epsilon/200$-close to connected and $\epsilon/200$-close to having an EMST-completion with a crossing-free straight line embedding (if this is not the case, the property testers for connectivity and crossing-free EMST-completions reject). This way we develop a property tester for graphs with maximum degree 5. The degree bound is then removed in Section 4.5.

4.4.1 Testing connectivity

In the first phase we test whether the input graph is connected. We say a geometric graph G for P is ϵ-*far from connected* if one has to add more than ϵn edges to G to obtain a connected graph. If G is not ϵ-far from connected, then we call it ϵ-*close* to connected.

Remark 1 *Let us notice here the equivalent characterization of geometric graphs for* P *that are* ϵ*-far from connected — these are geometric graphs for* P *having more than* $\epsilon n + 1$ *connected components.*

Since the property of being connected does not depend on the positions of the input points in P, we can use a property tester for connectivity in graphs.

Lemma 4.19 [13] *Let* G *be a graph with degree bound* d*. Connectivity of* G *can be tested with* $\mathcal{O}\left(\frac{\log^2(1/\epsilon d)}{\epsilon d}\right)$ *time and query complexity in the bounded length adjacency list model*[3].

We can immediately apply this result to geometric graphs:

Corollary 4.20 *Let* G *be a geometric graph for* P *with maximum degree 5. There is a property tester that in time* $\mathcal{O}\left(\frac{\log^2(1/\epsilon)}{\epsilon}\right)$ *and with a query complexity of* $\mathcal{O}\left(\frac{\log^2(1/\epsilon)}{\epsilon}\right)$ *accepts the input if* G *is connected and rejects the input with probability at least* $\frac{2}{3}$ *if* G *is* ϵ*-far from connected.*

Proof : We run the tester from [13] with $d = 5$ and $\epsilon' = \epsilon/5$. □

4.4.2 Testing crossing-free EMST-completions

Next, we design a property tester that accepts the input graph if it is the EMST and rejects it if the straight-line embedding of its EMST-completion is ϵ-far from crossing-free. We say the straight-line embedding of a geometric graph G for P is *ϵ-far from crossing-free* if one has to remove more than $\epsilon\, n$ edges in G to obtain a crossing-free straight-line embedding. If the straight-line embedding of G is not ϵ-far from crossing-free, then we call it *ϵ-close* to crossing-free.

We proceed in two steps. First, our property tester checks for pairs of intersecting blue edges and then for intersections between blue and red edges; red edges cannot intersect because they are edges of the EMST.

We first use the tester DISJOINTNESS developed in Section 3 to find intersections between blue segments (induced by blue edges). Since G has maximum degree 5 it has at most $2.5\,n$ edges. Therefore, since one can verify in time $\mathcal{O}(n \log n)$ if a geometric graph with n vertices has crossing-free straight-line embedding [1], Theorem 1 implies the following result.

Lemma 4.21 *Let* G *be a geometric graph with maximum degree 5. There is a property tester that in time* $\mathcal{O}(\sqrt{n/\epsilon}\log(n/\epsilon))$ *and with the query complexity of* $\mathcal{O}(\sqrt{n/\epsilon})$ *accepts the input if the straight-line embedding of* G *is crossing-free and rejects the input with probability at least* $\frac{2}{3}$ *if the straight-line embedding of* G *is* ϵ*-far from crossing-free.* □

It remains to design a property tester for red-blue intersections in the EMST-completion of G. (More precisely, we do not design a property tester for the property of having no red-blue intersections. Our algorithm might reject an input graph if its EMST-completion has no red-blue intersections. However, if the input graph is the EMST then it is always accepted by our algorithm.) A geometric graph with red and blue edges has a straight-line embedding without red-blue intersections if there is no intersection between the corresponding red and blue segments. Similarly, the straight-line embedding of a geometric graph whose edges are colored blue and red is *ϵ-far from having no red-blue intersections* if one has to delete more than an ϵ-fraction of its edges to remove all red-blue intersections.

The main difficulty with testing for red-blue intersections in the EMST-completion of G is caused by the fact that the red edges are defined only *implicitly*, because they do not belong to the input graph G. We will use the following lemma to study properties of intersections of explicitly given blue edges with implictly given red ones.

[3] In the bounded degree graph model a graph is ϵ-far from connected if one has to add more than $\epsilon\, d\, n$ edges to obtain a connected graph.

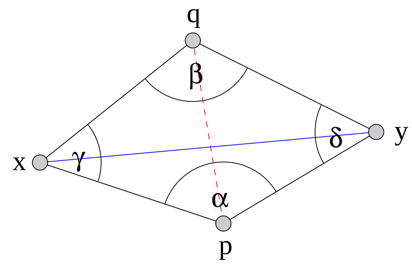

Figure 2: A quadrilateral $\langle p, q, x, y\rangle$ with red blue intersection. The red edge is dotted.

Lemma 4.22 *Let $\overline{pq}$ be a red and $\overline{xy}$ be a blue segment in the EMST-completion of G. If $\overline{pq}$ and $\overline{xy}$ intersect each other, then edge (x, y) either is not in the EMST of any set containing $\{x, y, p\}$ or (x, y) is not in the EMST of every set containing $\{x, y, q\}$.*

Proof : The points p, q, x, y are in convex position because the segments $\overline{pq}$ and $\overline{xy}$ intersect. We consider the quadrilateral $pxqy$ (see Figure 2). Let us call the inner angles in the quadrilateral at vertices p, q, x, y to be α, β, γ, and δ, respectively. Let us recall that the longest edge of a triangle is opposite of the largest angle.

If $\alpha < \frac{\pi}{2}$ and $\beta < \frac{\pi}{2}$ then γ or δ is larger than $\frac{\pi}{2}$ because $\alpha + \beta + \gamma + \delta = 2\pi$. Without loss of generality, let $\gamma > \frac{\pi}{2}$. Then, segment $\overline{pq}$ is the longest edge of triangle pqx and thus is cannot be the EMST of $\{p, q, x\}$. By Lemma 4.4 this is a contradiction to the fact that (p, q) is an edge of the EMST. Hence we must have either $\alpha \geq \frac{\pi}{2}$ or $\beta \geq \frac{\pi}{2}$.

If $\alpha \geq \frac{\pi}{2}$ then segment $\overline{xy}$ is the longest edge in triangle pxy. Hence edge (x, y) is not contained in the EMST of p, x, y. By Lemma 4.4 it is also not contained in any EMST of a subset of P that contains p, x, y. Similarly, if $\beta \geq \frac{\pi}{2}$ then edge (x, y) is not contained in any EMST of a subset of P that contains q, x, y. □

This lemma shows that each red-blue intersection between a red edge (p, q) and a blue edge (x, y) has a *"witness"* consisting of one point $z \in \{p, q\}$ and the edge (x, y) so that (x, y) is not in the EMST of x, y, z.

Our property tester for red-blue intersections is similar to the DISJOINTNESS tester but we use a modified definition of disjointness property: We say that *two points* $v, u \in P$ *intersect* if there is a point $w \in P$ such that at least one of (v, w) and (u, w) is a blue edge that is not in the EMST of v, u, w.

We now are ready to present our property tester for red-blue intersections:

REDBLUETEST(G, ϵ)
  Choose a set $S' \subset P$ of size $16\sqrt{5n/\epsilon}$ uniformly at random
  Let $S = S' \cup \mathcal{N}(S')$, where $\mathcal{N}(S')$ denotes the set of neighbors of points in S'
  Let G_S denote the subgraph induced by S
  if the EMST-completion of G_S has a cycle **then** *reject*
  else *accept*

The analysis of the algorithm is similar to the analysis of the algorithm DISJOINTNESS.

Lemma 4.23 *Let G be a geometric graph for P with maximum degree 5. Algorithm* REDBLUETEST *runs in time $\mathcal{O}(\sqrt{n/\epsilon}\log n)$ and with the query complexity of $\mathcal{O}(\sqrt{n/\epsilon})$, and accepts the input graph G if* *it is the EMST of P and rejects the input with probability at least $\frac{2}{3}$ if the straight-line embedding of the EMST-completion of G is ϵ-far from having no red-blue intersection.*

Proof : Obviously, if $G = (P, E)$ is the EMST then algorithm REDBLUETEST accepts G.

Let G_C denote the EMST-completion of G and let us assume that G_C is ϵ-far from having no red-blue intersections. By Lemma 4.22, we can apply the following procedure $k = \epsilon n/20$ times: pick a pair of intersecting (according to the definition above) points $\{v, u\} = W_i$, $i \in [k]$, and remove all edges incident to v and u from G_C. By the degree bound, we removed at most 10 edges for the two vertices, and therefore this procedure can be performed at least k times.

In order to prove that if G_C is ϵ-far from having no red-blue intersections then REDBLUETEST rejects G with probability at least $2/3$, we show first that with probability at least $2/3$ one of the sets W_i, $i \in [k]$, is in S'. We apply Lemma A.1 [7] to obtain:

$$\mathbf{Pr}\left[\exists j \in [k] : (W_j \subseteq S')\right] \;\geq\; 1 - (1 - 3/4)^4 \;\geq\; 2/3 \;.$$

It remains to show that if there is a set $W_i \subseteq S'$ then the algorithm rejects the input graph. If $W_i = \{v, u\} \subseteq S'$ then there exists a blue edge $e = (v, w)$ (or $e = (u, w)$) such that e is not in the EMST of v, u, w. Therefore, by Lemma 4.4, e is not in the EMST of S. Hence S has a cycle and REDBLUETEST rejects. □

Finally, we can combine Lemmas 4.21 and 4.23 to obtain the following result.

Lemma 4.24 *Let G be a geometric graph for P with maximum degree 5. There is an algorithm that in time $\mathcal{O}(\sqrt{n/\epsilon}\,\log(n/\epsilon))$ and with a query complexity of $\mathcal{O}(\sqrt{n/\epsilon})$ accepts G if G is the EMST of P and rejects G with probability at least $\frac{2}{3}$ if the straight-line embedding of the EMST-completion of G is ϵ-far from crossing-free.*

Proof : Let G be ϵ-far from having a crossing-free EMST-completion. Then, either the straight-line embedding of G is $\frac{\epsilon}{2}$-far from crossing-free or the EMST-completion of G is $\frac{\epsilon}{2}$-far from having no red-blue intersections. Applying Lemma 4.21 and Lemma 4.23 with $\epsilon = \frac{\epsilon}{2}$ shows that G is rejected with probability at least $\frac{2}{3}$. Since the tester for blue-blue intersection and the tester for red-blue intersections both accept the EMST, this completes the proof of Lemma 4.24. □

4.5 Property tester in general graphs

It remains to remove the degree bound condition for the input graph. We begin with a low degree tester.

4.5.1 Property tester for low degree

A geometric graph $G = (P, E)$ for P is ϵ-*far from having low degree* if one has to remove more than ϵn edges in G to obtain a graph having maximum degree smaller than or equal to five. If G is not ϵ-far from having low degree then we call it ϵ-*close* to having small degree.

It is easy to see that if G is ϵ-far from having small degree, then there are at least $\sqrt{\epsilon n}$ vertices in G either having degree greater than five or having an adjacent vertex with degree greater than five. Therefore the simple algorithm that picks a random set S of $4\sqrt{n/\epsilon}$ points in P and tests if every point $p \in S$ has the degree smaller than or equal to 5 and if so then it tests if every neighbor of $p \in S$ has degree smaller than or equal to 5, will detect with probability greater than or equal to $\frac{2}{3}$ every geometric graph G that is ϵ-far from having small degree.

Lemma 4.25 *Let* G *be a geometric graph for* P*. There is a property tester that in time* $\mathcal{O}(\sqrt{n/\epsilon})$ *and with the query complexity of* $\mathcal{O}(\sqrt{n/\epsilon})$ *accepts the input if* G *has the maximum degree smaller than or equal to* 5 *and rejects the input with probability at least* $\frac{2}{3}$ *if* G *is* ϵ*-far from having small degree.*

Proof : Clearly, our algorithm accepts every graph having maximum degree of five. Let us assume that G is ϵ-far from having small degree. A vertex that has degree more than five or that is adjacent to a vertex with degree more than 5 is called a *heavy vertex*. Let S denote a sample of size $4\sqrt{\epsilon\, n}$ chosen uniformly at random from P.

$$\mathbf{Pr}\,[\text{S contains no heavy vertex}] \;\leq\; (1 - 1/\sqrt{n/\epsilon})^{4\sqrt{n/\epsilon}} \;\leq\; 1/3 \enspace .$$

It follows that

$$\mathbf{Pr}\,[\text{S contains a heavy vertex}] \;\geq\; 2/3 \enspace .$$

Hence our algorithm rejects every graph that is ϵ-far from having small degree with probability at least $2/3$. Thus it is a property tester.

The running time and the query complexity follow from the fact that since the degree of every $p \in S$ is less than or equal to 5, all operations can be performed in a constant time per vertex p. □

4.5.2 Tester for general graphs

To obtain a tester for general graphs we modify the tester for graph with degree bound of 5 in the following way: We first test whether the input graph is $\epsilon/2$-far from having low degree (using the tester described below). Then we run the property tester for graphs with maximum degree of 5 after applying the following modifications and with distance parameter $\varepsilon = \epsilon/2$:

- If during the course of the algorithm we encounter a vertex with degree greater than 5, we immediately reject.
- For each vertex $v \in S$ we also include every neighbor of v in G into the sample set. This can be done without asymptotically increasing the running time of the algorithm (because if we encounter a vertex with degree greater than 5 then we reject).

Clearly, the above modifications do not affect the case when the input graph is the EMST of the point set: the algorithm will still accept the input graph. Thus let us consider the case when the input graph G is ϵ-far from EMST. If the low degree tester rejects the input graph, we are done. Thus let us assume that the input graph passes this test (and thus is $\epsilon/2$-close to having low degree) but is ϵ-far from EMST. We call a vertex of G a *distinguisher* if either it has degree greater than 5 or it has a neighbor whose degree is greater than 5. Now we define the graph G' to be a graph obtained from G by deleting a minimal set of edges such that G' has maximum degree of 5. Since we deleted less than $\epsilon\, n/2$ edges from G to obtain G', we conclude that G' is $\epsilon/2$-far from EMST.

In order to analyze the behavior of the modified algorithm we consider the (unmodified) algorithm for graphs with maximum degree 5. First of all, we observe that if there is a distinguisher in the sample chosen by the unmodified algorithm then the modified algorithm always rejects. But if there is no distinguisher in the sample chosen by the unmodified algorithm then the graph “looks” like the graph G' which has maximum degree 5 and is $\epsilon/2$-far from EMST. If we run the unmodified algorithm on input G' it rejects with probability $2/3$. Thus the modified algorithm always rejects when the unmodified algorithm rejects G'. We conclude that the modified algorithm rejects G with probability at least $2/3$ because it either rejects or it behaves like the unmodified algorithm with input G'. Hence we proved:

Theorem 2 *There is a property tester for the EMST property with a running time of* $\mathcal{O}(\sqrt{n/\epsilon}) \cdot \log^2(n/\epsilon))$ *and with a query complexity of* $\mathcal{O}(\sqrt{n/\epsilon} \cdot \log(n/\epsilon))$. □

References

[1] J. L. Bentley and T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computers*, C-28:643–647, 1979.

[2] B. Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. *Journal of the ACM*, 47(6): 1012–1027, November 2000.

[3] B. Chazelle, R. Rubinfeld, and L. Trevisan. Approximating the minimum spanning tree weight in sublinear time. In *Proceedings of the 28th Annual International Colloquium on Automata, Languages and Programming (ICALP)*, pages 190–200, 2001.

[4] A. Czumaj, F. Ergün, L. Fortnow, A. Magen, I. Newman, R. Rubinfeld, and C. Sohler. Sublinear-time approximation of Euclidean minimum spanning tree. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 813–822, 2003.

[5] A. Czumaj and C. Sohler. Estimating the weight of metric minimum spanning trees in sublinear-time. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 175–183, 2004.

[6] A. Czumaj, C. Sohler, and M. Ziegler. Property testing in computational geometry. In *Proceedings of the 8th Annual European Symposium on Algorithms (ESA)*, pp. 155–166, 2000.

[7] A. Czumaj, C. Sohler, M. Ziegler. Testing Convex Position. Manuscript, 2004.

[8] B. Dixon, M. Rauch, and R. E. Tarjan. Verification and sensitivity analysis of minimum spanning trees in linear time. *SIAM Journal on Computing*, 21(6):1184–1192, 1992.

[9] D. Eppstein. Spanning trees and spanners. In *Handbook of Computational Geometry*, pp. 425–461. Elsevier Science B.V., 1997.

[10] E. Fischer. The art of uniformed decisions. A primer to property testing. *Bulletin of the European Association for Theoretical Computer Science*, 75:97 – 126, 2001.

[11] O. Goldreich. Combinatorial property testing (a survey). In P. Pardalos, S. Rajasekaran, and J. Rolim, editors, *Proceedings of the DIMACS Workshop on Randomization Methods in Algorithm Design*, volume 43 of *DIMACS, Series in Discrete Mathematics and Theoretical Computer Science*, pages 45–59, 1997. American Mathematical Society, Providence, RI, 1999.

[12] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[13] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2): 302–343, 2002.

[14] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. *Random Structures and Algorithms*, 23(1): 23–57, August 2003.

[15] T. Hagerub and C. Rüb. A guided tour to Chernoff bounds. *Information Processing Letters*, 33:305–308, 1989.

[16] D. R. Karger, P. N. Klein, and R. E. Tarjan. A randomized linear-time algorithm to find minimum spanning trees. *Journal of the ACM*, 42(2): 321–328, March 1995.

[17] V. King. A simpler minimum spanning tree verification algorithm. *Algorithmica*, 18:263–270, 1997.

[18] D. G. Kirkpatrick and J. D. Radke. A framework for computational morphology. In *Computational Geometry*, pages 217–248, North-Holland, 1985.

[19] J. Komlós. Linear verification for spanning trees. *Combinatorica*, 5(1): 57–65, 1985.

[20] M. Parnas and D. Ron. Testing the diameter of graphs. *Random Structures and Algorithms*, 20(2): 165–183, March 2002.

[21] S. Pettie and V. Ramachandran. An optimal minimum spanning tree algorithm. *Journal of the ACM*, 49(1): 16–34, 2002.

[22] D. Ron. Property testing. In *Handobook of Randomized Algorithms*, volume II, pages 597–649, Kluwer Academic Publishers, 2001.

[23] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[24] M. I. Shamos and D. Hoey. Closest-point problems. In *Proceedings of the 16th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 151–162, 1975.

[25] R. E. Tarjan. Data structures and network algorithms. In *CBMS-NSF Regional Conference Series in Applied Mathematics*, volume 44. Society for Industrial and Applied Mathematics (SIAM), 1983.

Appendix

A An auxiliary sampling witnesses lemma

In the paper we are using the following lemma from [7](see also [6]) that shows that if we take a sample of size s from a set Ω of n elements, then with good probability we have at least one of k disjoint sets (each of size ℓ) in our sample. For the sake of completeness, we present the proof of this lemma in the current paper.

Lemma A.1 *Let Ω be an arbitrary set of n elements. Let k and l be arbitrary integers (possibly dependent on n) and let s be an arbitrary integer such that $s \ge \frac{2n}{(2k)^{1/\ell}}$. Let $W_1, \dots, W_k$ be arbitrary disjoint subsets of Ω each of size ℓ. Let S be a subset of Ω of size s which is chosen independently and uniformly at random. Then*

$$\mathbf{Pr}\left[\exists j \in [k] \ : \ (W_j \subseteq S)\right] \ge \frac{1}{4} \ .$$

Proof : We first observe that we can focus on the case $k \le \frac{n}{2}$, because if $k > \frac{n}{2}$, then every set W_i contains exactly one element and then we immediately get $\mathbf{Pr}[\exists j \in [k] \ : \ (W_j \subseteq S)] \ge \frac{1}{2}$. Furthermore, since $k \le \frac{n}{2}$ yields $\ell + \frac{n-\ell}{(2k)^{1/\ell}} \le \frac{2n}{(2k)^{1/\ell}}$, it is sufficient to consider only the case $s = \ell + \frac{n-\ell}{(2k)^{1/\ell}}$. Next, by Boole-Benferoni inequality we obtain

$$\mathbf{Pr}\left[\exists j \in [k] \ : \ (W_j \subseteq S)\right] \ \ge \ \sum_{j=1}^{k} \mathbf{Pr}[W_j \subseteq S] - \sum_{1 \le i < j \le k} \mathbf{Pr}[(W_i \cup W_j) \subseteq S] \ .$$

Furthermore, we observe that for every $j \in [k]$ it holds that

$$\mathbf{Pr}[W_j \subseteq S] \ = \ \frac{\binom{n-\ell}{s-\ell}}{\binom{n}{s}} \ = \ \frac{(n-\ell)!}{(s-\ell)!\,(n-s)!} \cdot \frac{s!\,(n-s)!}{n!} \ = \ \frac{(n-\ell)!\,s!}{n!\,(s-\ell)!} \ = \ \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} \ .$$

Similar arguments can be used to show that for every $i, j \in [k]$, if $i \neq j$, then it holds that

$$\mathbf{Pr}[(W_i \cup W_j) \subseteq S] \ = \ \frac{\binom{n-2\ell}{s-2\ell}}{\binom{n}{s}} \ = \ \prod_{r=0}^{2\ell-1} \frac{s-r}{n-r} \ = \ \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} \cdot \prod_{r=0}^{\ell-1} \frac{(s-\ell)-r}{(n-\ell)-r} \ .$$

Hence, Boole-Benferoni inequality implies that

$$\begin{aligned}
\mathbf{Pr}\left[\exists j \in [k] \ : \ (W_j \subseteq S)\right] \ &\ge \ k \cdot \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} - \binom{k}{2} \cdot \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} \cdot \prod_{r=0}^{\ell-1} \frac{(s-\ell)-r}{(n-\ell)-r} \\
&= \ k \cdot \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} \cdot \left(1 - \frac{k-1}{2} \cdot \prod_{r=0}^{\ell-1} \frac{(s-\ell)-r}{(n-\ell)-r}\right) \\
&\ge \ k \cdot \prod_{r=0}^{\ell-1} \frac{s-\ell}{n-\ell} \cdot \left(1 - k \cdot \prod_{r=0}^{\ell-1} \frac{s-\ell}{n-\ell}\right) \\
&= \ k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} \cdot \left(1 - k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell}\right) \ .
\end{aligned}$$

Now, since we assumed that $s = \ell + \frac{n-\ell}{(2k)^{1/\ell}}$, our calculations above yield $\mathbf{Pr}[\exists j \in [k] \ : \ (W_j \subseteq S)] \ge \frac{1}{4}$, and thus the lemma follows. □